

Extension de la métaphore collaborative et application à un environnement de création

Ali Allaoui

Laboratoire ICA - Acroe
46, avenue Felix Viallet
38000, Grenoble, France
ali.allaoui@imag.fr

RESUME

Cet article s'interroge sur la multimodalité dans les environnements interactifs de création. Une bimodalité particulière est étudiée : la collaboration manipulation directe / "manipulation textuelle". Cette étude est basée sur la notion de métaphore collaborative de Hutchin. Nous détaillons cette métaphore de manière à aboutir à une collaboration inédite entre les deux modalités. Cette collaboration est expliquée dans le cas du logiciel MIMESIS, environnement dédié à la modélisation physique particulière.

MOTS CLES : multimodalité, métaphore collaborative, environnement de création

ABSTRACT

This paper wonders about the multimodality in the interactive creation environments. A particular bimodality is studied : the collaboration between direct manipulation and "textual manipulation". This study is driven by the notion of collaborative metaphore, introduced by Hutchin. We will detail this metaphore so we can introduce a novel collaboration between these two modalities. The collaboration is explained in the case of MIMESIS software, an environment dedicated to physically-based particle modelling.

CATEGORIES AND SUBJECT DESCRIPTORS: H5.2.

GENERAL TERMS: Design.

KEYWORDS: multimodality, collaborative metaphore, creation environment

INTRODUCTION

D'un point de vue théorique, la multimodalité provoque trois questions : la sélection de l'objet d'intérêt, l'allocation d'une modalité, et la combinaison de différentes modalités. Dans le cadre des environnements de création, nous estimons que l'allocation d'une modalité, dans certains cas, doit être du ressort de l'utilisateur. Il faut alors assurer la combinaison des différentes modalités pouvant être utilisées pour une même tâche.

Nous nous concentrons sur la combinaison, et plus particulièrement la combinaison entre deux grandes familles de modalités, introduites par Frohlich dans [3] : la première basée sur la métaphore conversationnelle et la seconde sur la métaphore dite "Model-World". Ces deux métaphores, introduites par Hutchin dans [5], opposent les interfaces basées sur des interactions langagières aux interfaces basées sur l'action. Le paradigme de la manipulation directe, qui a connu un large succès depuis sa théorisation par Schneiderman [8], est au cœur de la métaphore "Model-World".

Hutchin [5], ainsi que Frohlich [4], ont pris soin de tempérer l'enthousiasme initial pour ce paradigme, sans toutefois opposer les partisans respectifs de la métaphore conversationnelle et de la métaphore "Model World". Hutchin propose ainsi la métaphore collaborative, dans laquelle "l'utilisateur doit pouvoir avoir une conversation sur le monde avec l'agent, et l'agent comme l'utilisateur doit pouvoir manipuler le monde partagé". Cette métaphore nous intéresse dans le cadre de MIMESIS [2], environnement de modélisation de réseaux masse-interaction pour l'image animée. En effet, selon le réseau à construire, l'utilisateur doit pouvoir choisir entre langage et action, tout en gardant la possibilité de mettre en interaction des sous-réseaux construits avec des moyens différents.

SCHEMA CLASSIQUE DE LA METAPHORE COLLABORATIVE

Hutchin résume le concept de métaphore collaborative à l'aide du schéma en figure 1.

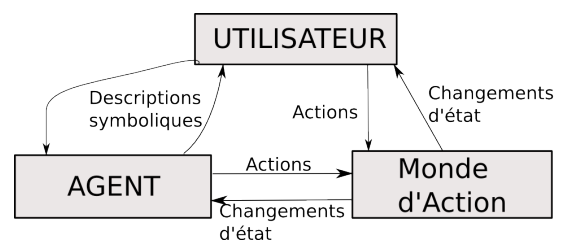


Figure 1 : Métaphore collaborative classique

L'agent est un widget avec lequel on interagit par le langage, le monde d'action est un ensemble de widgets avec lesquels on interagit par l'action. Ainsi, l'utilisateur peut interagir symboliquement avec un agent ou gestuellement avec le monde d'action. Ce choix lui permet de s'adapter à la tâche qu'il veut réaliser, selon que celle-ci requiert un haut niveau d'abstraction ou non.

Ce schéma a une portée très générale, il représente plus une famille de métaphores qu'une unique métaphore. Cette variété est déjà illustrée à travers les exemples donnés par Hutchin dans [3], mais il ne cherche pas à caractériser l'origine de cette variété. Frohlich tente d'aller plus loin dans [4] en s'appuyant sur le framework de [3] pour distinguer "separate mixed mode combinations" (combinaison entre langage et action en entrée ou bien en sortie) et "cross-modal combinations" (combinaison entre langage en entrée et action en sortie, ou l'inverse).

Nous nous proposons de préciser ce schéma en insistant sur les points suivants :

1. Place de l'objet d'intérêt
2. Le critère syntactique
3. Nature de l'agent :
 - a. Type de langage
 - b. Rôle, fonction
 - c. L'agent représente-t-il un objet ?
4. Intégration au monde d'action
5. Cardinalité

L'OBJET D'INTERET

La notion d'objet d'intérêt fait défaut à ce schéma classique. Il n'est présent qu'implicitement ce qui confère à ce schéma une première ambiguïté. La notion même d'objet d'intérêt prête à confusion. On peut le définir simplement comme l'ensemble des données que veut éditer l'utilisateur. Mais cette définition très générale ne lève pas la confusion entre représentant et représenté. L'objet d'intérêt est-il les données ou la représentation des données ? La confusion est d'autant plus facile que souvent cette représentation est l'objet final désiré par l'utilisateur (e.g., un document Open Office). Dans le cas d'un modèleur 3D, la nuance est mieux perceptible : on distingue la scène 3D, qui est modélisée par l'utilisateur mais qui n'est qu'information, du rendu, produit final, qui est une image. Assurément, l'objet d'intérêt est la scène 3D.

Cette mise au point est importante car elle permet de montrer comment l'agent et le monde d'action peuvent constituer des représentations différentes de l'objet d'intérêt, et donc des moyens d'actions différents sur l'objet d'intérêt. Ce rôle pivot de l'objet d'intérêt est également déterminant pour la collaboration entre agent et monde d'action.

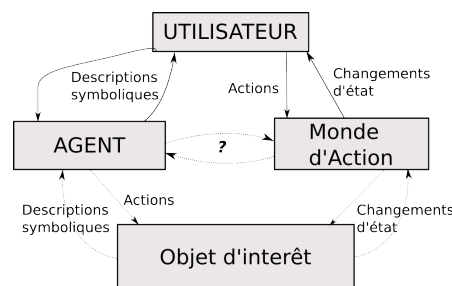


Figure 2 : Métaphore collaborative et objet d'intérêt

LE CRITERE SYNTACTIQUE

Le critère syntactique compare le pouvoir d'expression offert par l'agent à celui offert par le monde d'action. Ce critère est très semblable à l'aspect syntactique d'une combinaison de modalités introduit dans [9] par Vernier et Nigay. Notre critère est cependant différent car il compare ce que permet d'exprimer chacune des modalités, non pas ce qu'elle expose.

Le critère syntactique a deux dimensions, l'étendue et l'efficacité. L'étendue correspond à la relation entre l'espace des actions directes et l'espace des descriptions symboliques. Comme on peut le voir sur la figure 2, ces deux espaces sont liés à l'espace des actions admises par l'objet d'intérêt (une notification de changement d'état correspond à une action). Ces deux espaces ont-ils une relation d'équivalence, d'inclusion, d'inclusion partielle, ou d'exclusion ? Le pouvoir d'expression doit également être considéré par rapport à l'efficacité du langage d'interaction, on pense notamment aux possibilités offertes par les structures de contrôle (boucle for, while, condition, etc...) dans les descriptions symboliques. Cette efficacité ne peut être mesurée que par rapport à une tâche donnée.

DIVERSITE DES AGENTS

La nature de l'agent est déterminante dans la collaboration. Trois caractéristiques la définissent.

Type de langage de l'agent

Nous distinguons deux types de langage :

- Le langage impératif (e.g. : agent=terminal, langage = shell, objet d'intérêt=système)
- Le langage déclaratif (e.g. : agent = éditeur de texte, langage = html, objet d'intérêt = page web).

Les premiers insistent sur le processus de construction de l'objet, tandis que les deuxièmes insistent sur la structure de l'objet.

Rôle de l'agent

Nous proposons deux rôles que peut jouer un agent dans une interface :

1. Rôle d'assistant : l'agent aide l'utilisateur dans sa tâche. Il peut s'agir d'une aide simple, l'agent utilisant sa connaissance des objets et du contexte pour donner des choix à l'utilisateur. Mais il peut aussi s'agir d'une aide experte, où l'agent observe les manipulations de l'utilisateur et l'aiguille si besoin.
2. Rôle d'ouvrier : l'agent construit et/ou édite un objet d'intérêt sur instructions de l'utilisateur.

Agent représentation ?

L'agent peut constituer une représentation de :

- L'objet d'intérêt
- Une partie de l'objet d'intérêt
- Un objet auxiliaire, assistant l'utilisateur dans sa tâche
- Aucun objet

INTEGRATION DE L'AGENT AU MONDE D'ACTION

Ce critère est semblable à l'aspect spatial d'une combinaison décrit dans [9]. Nous distinguons 4 types d'intégration :

1. Séparation du monde d'action (Editeur de texte, ou terminal pour ligne de commande)
2. Objet mixte (action + texte) : agent et objet d'action forment une entité à part entière (e.g. , un fichier dans un explorateur de fichiers).
3. Attaché à objet d'action. L'agent attaché intervient quand on manipule l'objet d'action de manière équivoque (e.g., menu contextuel suite à un glisser/déplacer sur un fichier).
4. Plongé dans le monde d'action : l'agent est intégré au monde d'action mais ne possède pas tous les attributs d'un objet d'action (e.g. une fenêtre de chat dans un monde virtuel 3D, on ne peut que écrire dedans, pas la bouger ni la toucher)

Notons que le deuxième type est proche de la notion de modalité composite introduite par Bernsen dans [1]. On remarque également que ce critère d'intégration n'est pas uniquement spatial, il qualifie de manière plus large la relation de l'agent avec le monde d'action.

FAMILLE DES METAPHORES COLLABORATIVES MIXTES

Nous sommes à présent en mesure de dériver le concept de métaphore collaborative vers ce qu'on pourrait appeler métaphore collaborative mixte, dont la propriété constitutive est une intégration autre que séparé du monde d'action.

Cette famille recouvre notamment ce que Frohlich avait qualifié de fausses interfaces à manipulation directe (bureau Mac, Explorateur Windows) dans le sens où elles faisaient intervenir le langage dans la manipulation. Nous nous intéressons à un autre pan de cette famille,

basée sur la décomposition de l'objet d'intérêt en sous-objets.

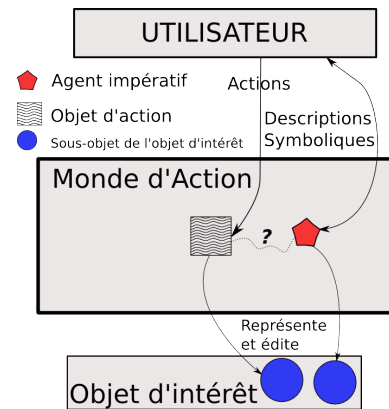


Figure 3 : Métaphore collaborative mixte et décomposition de l'objet d'intérêt

Ce schéma illustre la possibilité pour un sous-objet d'être représenté et édité soit par un objet d'action soit par un agent (ici impératif, mais il pourrait être déclaratif). Dans ce cas, plusieurs agents peuvent coexister, chacun représentant un sous-objet différent. On remarque un lien entre l'objet d'action et l'agent, qui représente un lien entre les deux sous-objets. Deux questions sont alors soulevées :

1. Quelle est la nature du lien entre les sous-objets ?
2. Comment l'utilisateur peut-il créer et éditer ce lien ?

Pour mieux comprendre ces deux questions, étudions les dans le cadre de MIMESIS.

METAPHORE COLLABORATIVE DANS MIMESIS

L'objet d'intérêt est le réseau de masse-interaction, les sous-objets sont des sous-réseaux. Le monde d'action est un établi virtuel sur lequel l'utilisateur construit le réseau. Les objets d'action sont les représentations graphiques des différents modules MAT (modules masse ponctuelles) et LIA (modules interaction). Dans le formalisme CORDIS-ANIMA, langage de modélisation de réseaux de masse-interaction, les modules MAT possèdent un point de connexion, appelée point M, et les modules LIA deux points de connexion, appelés points L. Un point M peut être connecté à un nombre arbitraire de points L, un point L doit être connecté à un seul point M. L'établi permet de disposer des modules MAT et LIA et de les connecter entre eux. L'agent impératif est à définir, voyons le dans un premier temps comme une simple fenêtre d'édition de texte plongée dans le monde d'action.

Le lien entre les sous-réseaux peut être de deux natures :

1. Une connexion entre un point M et un point L

2. Une relation de copie : un sous-réseau représenté par une première modalité contient une ou plusieurs copies d'un sous-réseau représenté par une deuxième modalité.

Comment créer et éditer ce lien ? Plusieurs cas de figures doivent être analysés :

1. Connexion point M / point L par le langage. L'agent contenant le sous-réseau à connecter, ou un agent tierce, effectue cette connexion. Il doit connaître les noms des modules à connecter.
2. Connexion point M / point L par l'action. L'agent doit acquérir les capacités de connexion des objets d'action :
 - a. La voie la plus simple est donc de transformer la vue texte en vue graphique représentant les modules créés comme des objets d'action.
 - b. Il est aussi envisageable de créer des "macro-points de connexion", ce qui serait adapté à des sous-réseaux contenant un grand nombre de modules qu'il serait lourd et inutile (dans certains cas) d'afficher.
3. Copie par le langage. Un agent crée une copie d'un sous-réseau créé graphiquement, il suffit que l'agent puisse référencer tous les modules du sous-réseau.
4. Copie par l'action. L'action copier/coller effectué sur un agent plongé dans le monde d'action doit être précisé par l'utilisateur : copie sous forme d'agent ou sous forme d'objets d'action.

En conclusion, Quelle que soit la nature du lien entre les sous-objets, on retient deux cas :

1. La liaison est effectuée par le langage, ce qui requiert que les sous-objets reliés soient nommés.
2. La liaison est effectuée par l'action, ce qui requiert que l'agent puisse acquérir toute ou partie des propriétés des objets d'action.

Dans ce dernier cas, il faudra veiller à ce que l'agent ne soit pas altéré par sa transformation en objet d'action. On tient en effet à ce que l'agent représente de manière " parlante " le sous-réseau, le script contenu par l'agent ne doit donc pas se résumer à un historique des manipulations de l'utilisateur.

Ces différents liens peuvent être vus comme des moyens d'opérer un *transfert* sur des sous-objets d'intérêt, et donc à une *complémentarité* sur l'objet d'intérêt. Le transfert et la complémentarité sont deux des six primitives proposées par Martin dans [7] pour la caractérisation des interfaces multimodales.

CONCLUSION

Nous avons enrichi la métaphore collaborative de Hutchin et montré que cette enrichissement était générateur en termes de conception d'interfaces mixtes. L'utilisateur bénéficie d'une interface lui permettant de choisir la représentation qui convient pour chaque sous-partie de son modèle, et de relier ces sous-parties avec l'interaction appropriée. Cette métaphore, déjà esquissée dans KPovModeler [6], pourrait être utilisée dans d'autres logiciels de modélisation (Simulink, Maya) qui sont basés sur une métaphore collaborative moins flexible.

BIBLIOGRAPHIE

1. N.O. Bernsen. Foundations of multimodal representations: A taxonomy of representational modalities. *Interacting with Computers*, 6(4):347–371, 1994.
2. M. Evrard, A. Luciani, N. Castagné. MIMESIS : Interactive Interface for Mass-Interaction Modeling, In *Proceedings of CASA 2006*, Geneva, July 2006, Nadia Magnenat-Thalmann & al. editors. pp177–186.
3. D. Frohlich. The design space of interfaces. In Lars Kjeldahl, editor, *Multimedia - Principles, Systems, and Applications*, chapter 53-69. Springer-Verlag, Berlin, 1991.
4. D. Frohlich. Direct manipulation and other lessons. In M.G. Hellander, T.K. Landauer, and P.V. Prabhu, editors, *Handbook of HCI : Second completely revised edition*, chapter 22, pages 463–488. Elsevier Science, Amsterdam, 1996.
5. E. Hutchin. Metaphors for interface design. In M.M. Taylor, F. Neel, and D.G. Bouwhuis, editors, *The structure of multimodal dialogue*, pages 11–28. Amsterdam, 1988. Palanque, Ph., Paternò, F. (eds.). *Formal Methods in Human-Computer Interaction*. Springer-Verlag, Berlin, 1997.
6. <http://www.kpovmodeler.org/fr/index.html>
7. J.C. Martin. Tycoon: theoretical and software tools for multimodal interfaces intelligence and multimodality in multimedia interfaces. aaai press. In John Lee (Ed.), editor, *AAAI Proceedings*, 1998..
8. B. Shneiderman. The future of interactive systems and the emergence of direct manipulation. In *Proc. of the NYU symposium on user interfaces on Human factors and interactive computer systems*, pages 1–28, Norwood, NJ, USA, 1984. Ablex Publishing Corp
9. F. Vernier and L. Nigay. A framework for the combination and characterization of output modalities. In *DSV-IS2000, Limerick (IR), Lecture Notes in Computer Sciences, Springer-Verlag publ*, pages 32–48, 2000.